Computer Science Department

Stanford University

Comprehensive Examination in Software Systems

Fall 1998

READ THIS FIRST!!!

1. You should write your answers for this part of the Comprehensive Examination in a BLUE BOOK. Be sure to write your MAGIC NUMBER on the cover of every blue book that you use.
2. The number of POINTS for each problem indicates how elaborate an answer is expected. For example, a question worth 6 points or less doesn't deserve an extremely detailed answer, even if you feel you could expound at length upon it. Short, bulleted answers are encouraged.
3. The total number of points is 30, although you have one hour in which to take the exam.
4. This exam is CLOSED BOOK. You may NOT use notes, books, computers, other people, etc.
5. Show your work, since PARTIAL CREDIT will be given for incomplete answers. For example, you can get credit for making a reasonable start on a problem even if the idea doesn't work out. You can also get credit for realizing that certain approaches are incorrect.
6. If you are convinced you need to make an assumption to answer a question, state your assumptions(s) as well as your answer.
7. Be sure to provide justification for your answers.

Fall 1998 Comprehensive Exam: Software Systems (30 points total)

1. (10 points) In UNIX the file directory structure (including hard and soft links) may be an acyclic graph. In MS-DOS it may only be a tree structure.

   a. Compared to a tree structure, what are the advantages and disadvantages of the acyclic graph directory structure?

   b. Why is a general graph directory structure not usually used for a file system?

2. (10 points) This question concerns page replacement.

   a. When is a process considered to be thrashing?

   b. What is the difference between global and local page replacement algorithms?

   c. What are the advantages and disadvantages of the two types of page replacement algorithms?

3. (10 points) Let's assume that a new type of main memory is invented that is cheap and usually very fast -- almost as fast as hardware registers. The only problem is that 0.1% of the time you access this memory, it is actually very slow (as slow as a disk access). The problem is that you have no way of predicting ahead of time when these slow accesses will occur.

   a. Would you incorporate this new type of memory into your system? Why or why not?

   b. Assuming you must incorporate it, would you make any changes to the virtual memory system or the file system or the network software or the process scheduler or the structure of the kernel?